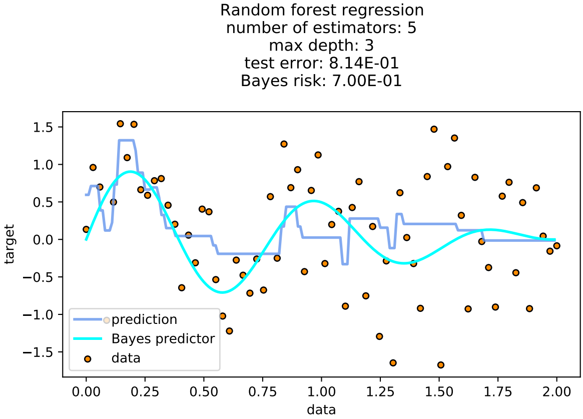
Fondamentaux théoriques du machine learning



Random forest regression
number of estimators: 5
max depth: 3
test error: 8.14E-01
Bayes risk: 7.00E-01

Ensemble learning

- *Bagging* and *boosting* are methods that combine estimators (in parallel or sequentially)
- they are often applied to decision trees
- they reach state-of-the-art performance in several supervised learning tasks [Fernández-Delgado et al., 2014]
- they are an active area of research (both theoretically and for practical applications).

`https://scikit-learn.org/stable/modules/ensemble.html`

Aggregating to reduce the variance

- usual setup : input space $\mathcal{X}$, output space $\mathcal{Y}$.
- we note z_b a dataset sampled from the unknown distribution ρ. If we sample b different datasets, $b \in [1, B]$,

$$z_b = \{(x_{1b}, y_{1b}), \dots (x_{nb}, y_{nb})\} \tag{1}$$

- we note $\hat{f}_{z_b}$ the estimator obtained after learning with z_b.

Aggregate to reduce the variance

- we note z_b a dataset sampled from the unknown distribution ρ. If we sample b different datasets, $b \in [1, B]$,

$$z_b = \{(x_{1b}, y_{1b}), \dots (x_{nb}, y_{nb})\} \tag{2}$$

- we note $\hat{f}_{z_b}$ the estimator obtained after learning with z_b.

Aggregating consists in using as an estimator :

- for regression

$$\hat{f}_B = \frac{1}{B} \sum_{b=1}^{B} \hat{f}_{z_b} \tag{3}$$

- for classification

$$\hat{f}_B(x) = \arg\max_j |\{b, \hat{f}_{z_b}(x) = j\}| \tag{4}$$

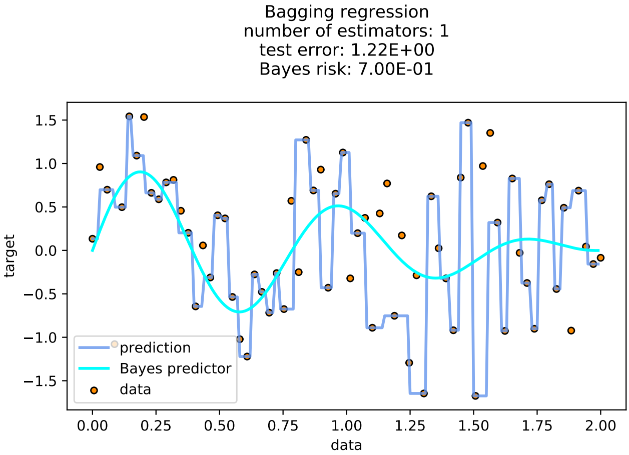
Bootstrapping and bagging

If B is large, it is not possible to sample B independent datasets with n samples, from a finite dataset.
Bootstrapping consists in sampling B times a sample dataset with n elements **with replacement**.
Bagging is the combination of bootstrapping and aggregating.

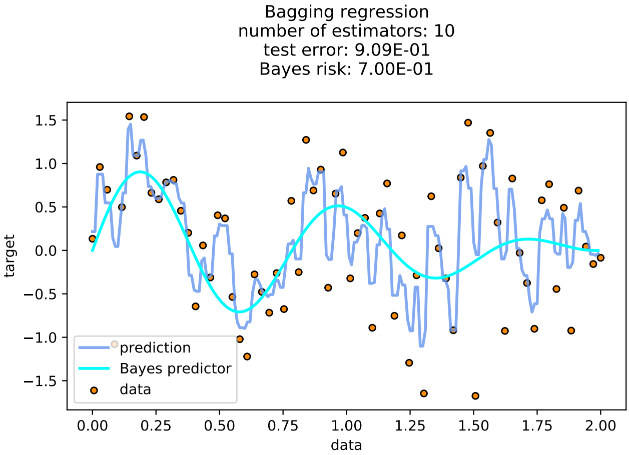
Out-of-bag error

Vocabulary : for an observation (x_i, y_i), the **out-of-bag error** is the mean error on this observation among the estimators that were trained on a bootstrap dataset **not** containing it.

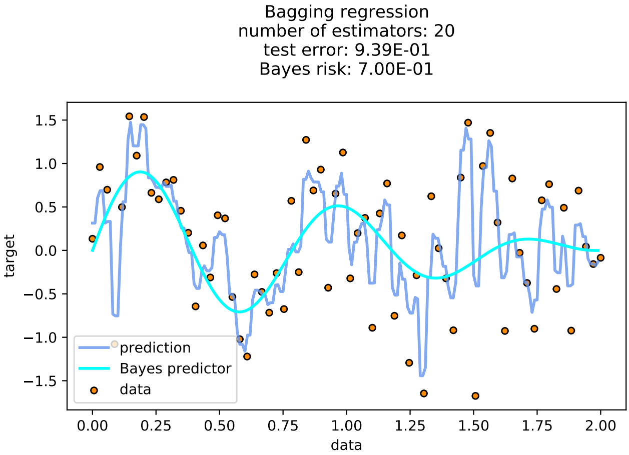
Simulation


Bagging regression
number of estimators: 1
test error: 1.22E+00
Bayes risk: 7.00E-01
target
data
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
prediction
Bayes predictor
data

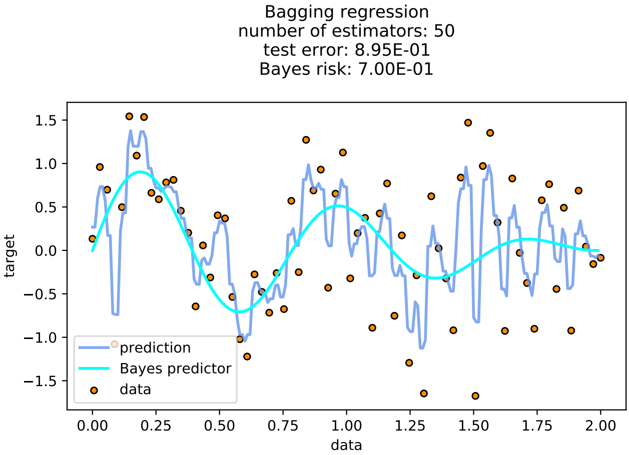
Simulation


Bagging regression
number of estimators: 10
test error: 9.09E-01
Bayes risk: 7.00E-01
target
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
data
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
prediction
Bayes predictor
data

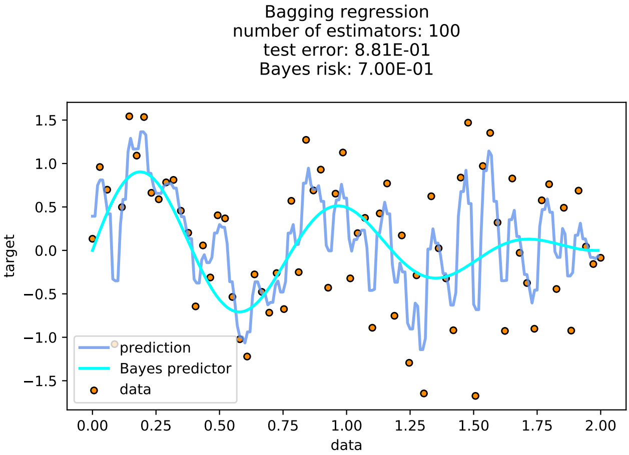
Simulation


Bagging regression
number of estimators: 20
test error: 9.39E-01
Bayes risk: 7.00E-01
target
data
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
prediction
Bayes predictor
data

Simulation


Bagging regression
number of estimators: 50
test error: 8.95E-01
Bayes risk: 7.00E-01
target
data
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
prediction
Bayes predictor
data

Simulation


Bagging regression
number of estimators: 100
test error: 8.81E-01
Bayes risk: 7.00E-01
target
data
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
prediction
Bayes predictor
data

Individual estimators

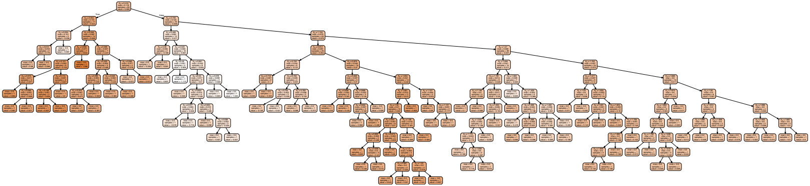


Figure – Estimator used in the averaging

Individual estimators

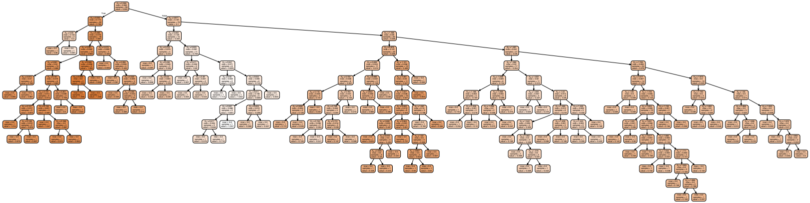


Figure – Other estimator used in the averaging

Possible issues

- number of trees to compute (B) in order to have a stable out-of-bag error.
- need for storing all the base estimators
- the final estimator is a *black-box* model.

Random forest

- Random forest [Breiman, 2001] is a bagging method with binary CART estimators, with additional randomness added to the choice of segmentation variables.
- the goal is to increase the **independence** between the base trees.
- although the theoretical properties of random forest are not yet fully understood, it is an important benchmark in supervised learning (but for instance not when the problem can be solved in a linear way).

Variance of an average : independent variables

Recall that if $(X_k)_{k\in\mathbb{N}}$ is a sequence of i.i.d. real variables that have a moment of order 2, and hence a variance σ^2, and an expected value of m. Then if $S_B = \frac{1}{B}\sum_{i=1}^{B} X_i$

$$
\begin{aligned}
Var(S_B) &= \sum_{i=1}^{B} Var(\frac{1}{B}X_i) \\
&= \frac{1}{B^2}\sum_{i=1}^{B} Var(X_i) \\
&= \frac{\sigma^2}{B}
\end{aligned}
\tag{5}
$$

Variance of the average : correlated variables

However, if the X_i are identically distributed but correlated with correlation c, then we admit that

$$Vac(S_B) = c\sigma^2 + \frac{1-c}{B}\sigma^2 \tag{6}$$

Random forest diminishes c by adding some randomness in the choice of the segmentation variables.

Random forest

```
Let $d$ be the number of features. Let $m \leq d$ be an integer.
Result: Random forest estimator
for $b \in [1, B]$ do
    Sample a bootstrap dataset $z_b$;
    Estimate $\hat{f_{z_b}}$ with randomization of the variables. The
     search of the optimal segmentation is done on $m$ randomly
     sampled variables.;
end
```

$$\hat{f_B} = \frac{1}{B} \sum_{b=1}^{B} \hat{f_{z_b}} \tag{7}$$

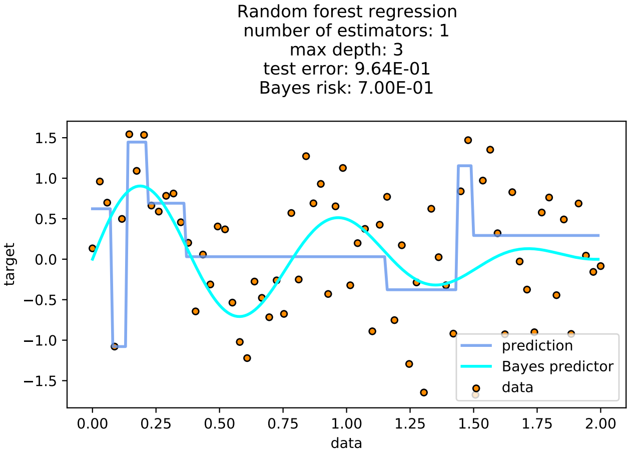
```
return $\hat{f_B}$
```

Algorithm 1: Random forest

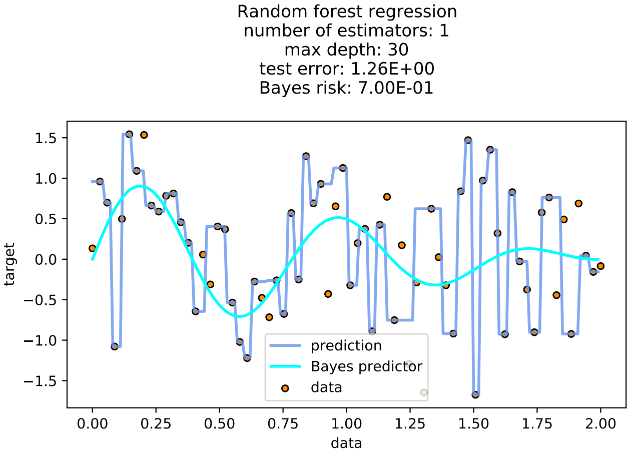
Remarks on random forest

- With random forest, it is possible to use a more brutal pruning (for instance using only trees of depth 2)
- To tune m, heuristics are used. Common ones include :
  - $m = \sqrt{d}$ for classification (d is the number of features)
  - $m = d/3$ for regression
  - cross validation.

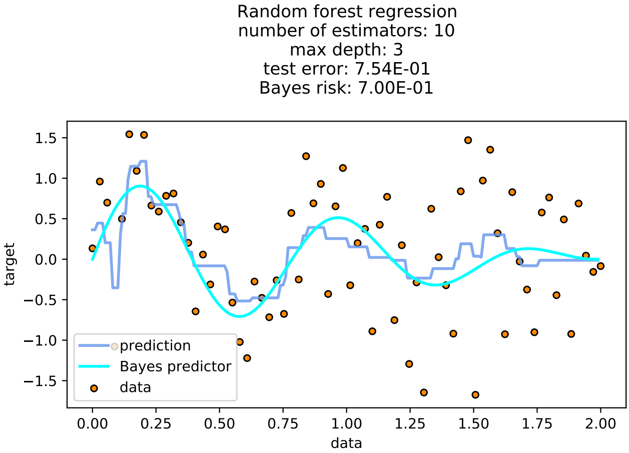
Example in 1D



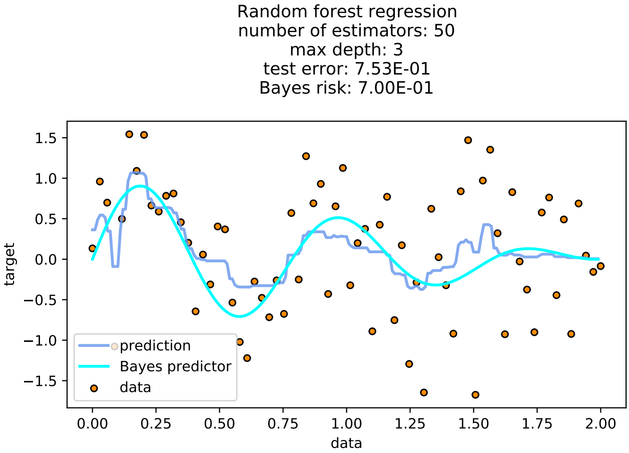
Example in 1D



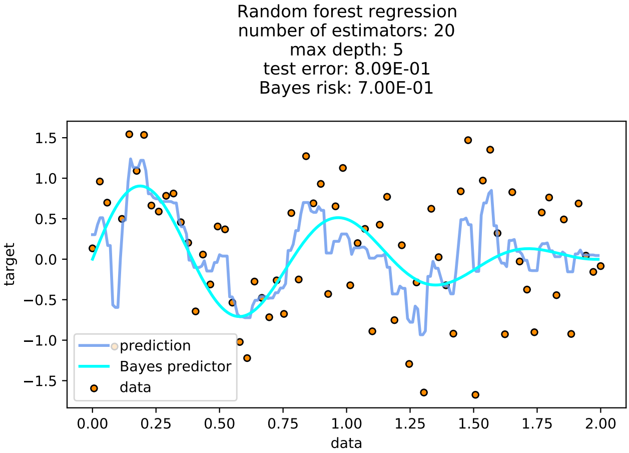
Example in 1D



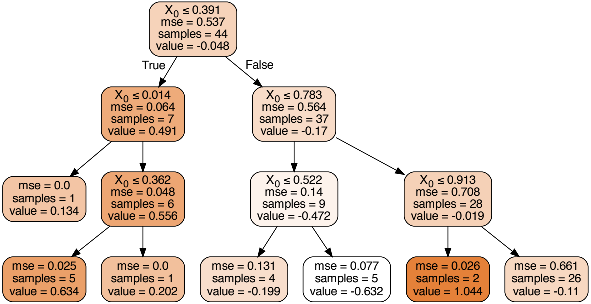
Example in 1D



Example in 1D



Estimators


$X_0 \leq 0.391$
mse = 0.537
samples = 44
value = -0.048
True
False
$X_0 \leq 0.014$
mse = 0.064
samples = 7
value = 0.491
$X_0 \leq 0.783$
mse = 0.564
samples = 37
value = -0.17
mse = 0.0
samples = 1
value = 0.134
$X_0 \leq 0.362$
mse = 0.048
samples = 6
value = 0.556
$X_0 \leq 0.522$
mse = 0.14
samples = 9
value = -0.472
$X_0 \leq 0.913$
mse = 0.708
samples = 28
value = -0.019
mse = 0.025
samples = 5
value = 0.634
mse = 0.0
samples = 1
value = 0.202
mse = 0.131
samples = 4
value = -0.199
mse = 0.077
samples = 5
value = -0.632
mse = 0.026
samples = 2
value = 1.044
mse = 0.661
samples = 26
value = -0.11

Importance of the variables

Although the obtained estimator is hard to interpret, we can still study the statistical importance of variables for the prediction.

Mean decrease accuracy

Consider a variable j. For a given tree b

- compute the out-of-bag error.
- shuffle the values of j in the out-of-bag sample.
- compute the new out-of-bag error
- store the decrease in prediction quality, D_j^b.

Average the D_j^b on b in order to measure the importance of the variable j.

Boosting

- While bagging is a random method, **boosting** is an adaptive method.
- Boosting builds on *weak classifiers*, and like bagging, aggregates them, for instance with a **weighted sum**.
- However, the weak classifiers are built **sequentially**. The classifier $b+1$ adapts the classifier b by focusing on improving the prediction of incorrectly classified training samples.
- Several variants exists (in the weighting, loss function, aggregating method, etc).

Adaboost

- Original boosting algorithm [Freund and Schapire, 1996]
- $\mathcal{Y} = \{-1, 1\}$ (but can also be adapted to a regression problem)
- Given the sample dataset $z = \{(x_1, y_1), \dots (x_n, y_n)\}$. We learn a **sequence of estimators** $(\delta_m)_{m \in [1..B]}$ (often CART).

```
Initialize the weights $w = \{w_i = \frac{1}{n}, i = 1..n\}$;
for $m = 1..B$ do
    Estimate $\delta_m$ on the weighted dataset. Compute the error rate
        $\hat{\epsilon_m} = \frac{\sum_{i=1}^{n} w_i 1(\delta_m(x_i) \neq y_i)}{\sum_{i=1}^{n} w_i}$    (8)
    Compute the logit of $\hat{\epsilon_m}$
        $c_m = \log\big(\frac{1 - \hat{\epsilon_m}}{\hat{\epsilon_m}}\big)$    (9)
    Update the weights
        $w_i \leftarrow w_i \exp\Big( - c_m 1(\delta_m(x_i) \neq y_i)\Big)$    (10)
end
$\hat{\delta_B} = \mathrm{sign}\Big(\sum_{b=1}^{B} \delta_{z_b}\Big)$    (11)
return $\hat{f_B}$
```

Algorithm 2: Adaboost for binary classification (adaptive boos-

Remarks

We need to enforce that $c_m \geq 0$. It is verified if $\hat{\epsilon_m} \leq \frac{1}{2}$.

$$c_m = \log\big(\frac{1-\hat{\epsilon_m}}{\hat{\epsilon_m}}\big) \tag{12}$$

- It is experimentally shown that if the weak classifiers are *stump trees* (2 leaves), then adaBoost performs better than one given tree with a number of leaves equal to the number of iterations of adaBoost (hence a comparable computation time).
- A number of leaves q between 4 and 8 for the weak classifiers is often recommended.
- adaBoost can be adapted to multiclass classification and regression [Schapire, 2003].

Gradient tree boosting

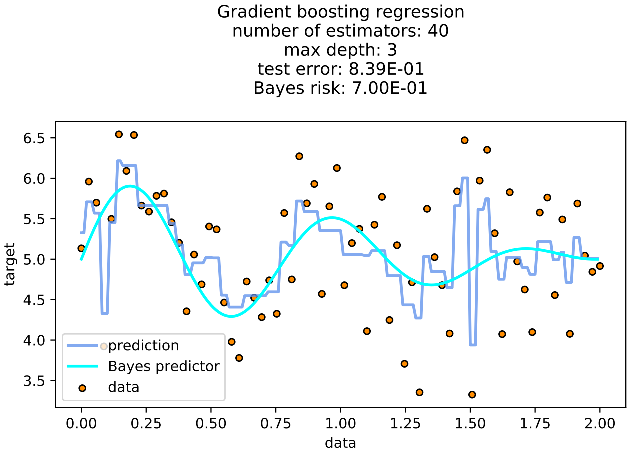
- differentiate with respect to the predictions of the tree (compute a gradient)
- approximate the gradient by a regression tree
- update the tree following this gradient.

Gradient tree boosting

```
Initialize $f_0 = \arg\min_z \sum_{i=1}^{n} l(y_i, z)$;
for $m = 1..B$ do
    Compute $r_{m_i} = -\frac{\partial l(y_i, f(x_i))}{\partial f(x_i)}(f = f_{m-1}), i \in [1, n]$
    Train a decision tree $\delta_m$ on $(x_i, r_{m_i})_{i \in [1,n]}$
    Compute $\gamma_m$ minimizing
        $$\gamma \to \sum_{i=1}^{n} l(y_i, f_{m-1}(x) + \gamma \delta_m(x_i)) \tag{13}$$
    Update the estimator
        $$\hat{f_m} = \hat{f_{m-1}} + \gamma_m \delta_m \tag{14}$$
end
return $\hat{f_B}$
```

Algorithm 3: Gradient tree boosting

Simulation


Gradient boosting regression
number of estimators: 40
max depth: 3
test error: 8.39E-01
Bayes risk: 7.00E-01
target
6.5
6.0
5.5
5.0
4.5
4.0
3.5
data
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
prediction
Bayes predictor
data

Gradient tree boosting

Many parameter can be tuned :

- maximum depth of the estimators
- shrinkage η ($\hat{f_m} = \hat{f_{m-1}} + \eta\gamma_m\delta_m$, with $0 \leq \eta \leq 1$). If η is low, e.g. ≤ 0.1, it can lead to a slower convergence but might prevent overfitting.
- number of trees learned B

`https://www.analyticsvidhya.com/blog/2016/02/`
`complete-guide-parameter-tuning-gradient-boosting-gbm-pytho`

It is necessary to experiment, read documentations, cross validate, use heuristics, etc.

Extreme gradient boosting

- XGBoost : variant of gradient boosting, very efficient on several benchmarks [Chen and Guestrin, 2016]
- can exploit parallelization
- `https://xgboost.readthedocs.io/en/latest/parameter.html`
- `https://github.com/dmlc/xgboost`

Catboost

- See also : Catboost `https://catboost.ai/`

Ensemble learning

- Bagging
- Random forest
- Boosting

Statistical learning

- Bounding the estimation error
- Interpolation regime and double descent

Model selection and sparsity

- Model selection
- Lasso

Statistical learning

- We come back to the statistical analysis of supervised learning.
- More precisely, to that of empirical risk minimization.

Reminder on risks

Let l be a loss. Generalization error :

$$R(f) = E_{(X,Y)\sim\rho}[l(Y, f(X))] \tag{15}$$

The **empirical risk (ER)** of an estimator f writes

$$R_n(f) = \frac{1}{n}\sum_{i=1}^{n} l(y_i, f(x_i)) \tag{16}$$

Remember that the risks depends on the loss l.

Risk decomposition

- f^* : Bayes predictor
- F : Hypothesis space
- f_n : estimated predictor (hence in F).

$$E\Big[R(f_n)\Big] - R^* = \Big(E\Big[R(f_n)\Big] - \inf_{f \in F} R(f)\Big) + \Big(\inf_{f \in F} R(f) - R^*\Big) \tag{17}$$

Risk decomposition

- f^* : Bayes predictor
- F : Hypothesis space
- f_n : estimated predictor.

When doing empirical risk minimization, f_n is obtained by minimization of the empirical risk.

However :

- we have seen that in many cases, finding the exact minimizer of the empirical risk might be computationnaly hard.
- also, a natural question is whether it is sufficient to have an **approximate minimizer** of the empirical risk, as the empirical risk is an **approximation** of the generalization error.

Risk decomposition

Estimation error (variance term, fluctuation error, stochastic error) : depends on D_n, F, f_n.

$$E\Big[R(f_n)\Big] - \inf_{f \in F} R(f) \geq 0$$

Approximation error (bias term) : depends on f^* and F, not on f_n, D_n.

$$\inf_{f \in F} R(f) - R^* \geq 0$$

It is also possible to consider the **Optimization error** : depends on D_n, F, f_n.

$$E\Big[R(\hat{f}_n) - R(f_n)\Big] \tag{18}$$

where $\hat{f}_n$ is an approximate solution to the optimization problem.

Bound on the estimation error

We will now focus on the estimation error

$$E\Big[R(f_n)\Big] - \inf_{f \in F} R(f) \geq 0$$

f_n is the empirical risk minimizer
We consider the best estimator in hypothesis space

$$f_a = \arg\min_{h \in F} R(h)$$

We have seen that

$$R(f_n) - R(f_a) \leq 2 \sup_{h \in F} |R(h) - R_n(h)| \tag{19}$$

Deterministic bound on the estimation error

$$f_a = \arg\min_{h \in F} R(h) \tag{20}$$

$$f_n = \arg\min_{h \in F} R_n(h) \tag{21}$$

$$\begin{aligned} R(f_n) - R(f_a) = &\left(R(f_n) - R_n(f_n)\right) \\ &+ \left(R_n(f_n) - R_n(f_a)\right) \\ &+ \left(R_n(f_a) - R(f_a)\right) \end{aligned} \tag{22}$$

Deterministic bound on the estimation error

$$
\begin{aligned}
f_a &= \arg\min_{h \in F} R(h) \\
f_n &= \arg\min_{h \in F} R_n(h) \qquad (23)
\end{aligned}
$$

$$
\begin{aligned}
R(f_n) - R(f_a) = &\big(R(f_n) - R_n(f_n)\big) \\
&+ \big(R_n(f_n) - R_n(f_a)\big) \qquad (24) \\
&+ \big(R_n(f_a) - R(f_a)\big)
\end{aligned}
$$

But by definition f_n minimizes R_n, so $\big(R_n(f_n) - R_n(f_a)\big) \leq 0$.

Deterministic bound on the estimation error

$$f_a = \arg\min_{h \in F} R(h)$$

$$f_n = \arg\min_{h \in F} R_n(h) \tag{25}$$

$$\begin{aligned} R(f_n) - R(f_a) = & \big(R(f_n) - R_n(f_n)\big) \\ & + \big(R_n(f_n) - R_n(f_a)\big) \\ & + \big(R_n(f_a) - R(f_a)\big) \end{aligned} \tag{26}$$

But by definition f_n minimizes R_n, so $\big(R_n(f_n) - R_n(f_a)\big) \leq 0$.
Finally :

$$R(f_n) - R(f_a) \leq 2 \sup_{h \in F} |R(h) - R_n(h)| \tag{27}$$

Bound on the estimation error

If we are able to bound $\sup_{h \in F} |R(h) - R_n(h)|$, then we have a bound on the estimation error.

Bound on the estimation error

Theorem

Weak law of large numbers
Let $(X_i)_{i\in\mathbb{N}}$ be a sequence of i.i.d. variables that have a moment of order 2. We note m their expected value. Then

$$\forall\epsilon > 0, \lim_{n\to+\infty} P\Big(|\frac{1}{n}\sum_{i=1}^{n} X_i - m| \geq \epsilon\Big) = 0 \tag{28}$$

We say that we have ***convergence in probability****.*

Bound on the estimation error

Theorem

Weak law of large numbers
Let $(X_i)_{i \in \mathbb{N}}$ be a sequence of i.i.d. variables that have a moment of order 2. We note m their expected value. Then

$$\forall \epsilon > 0, \lim_{n \to +\infty} P\big(|\frac{1}{n}\sum_{i=1}^{n} X_i - m| \geq \epsilon\big) = 0 \tag{29}$$

We say that we have ***convergence in probability****.*

However, this is only an asymptotical result : it is a limit for $n \to +\infty$.

Bound on the estimation error

If we are able to bound $\sup_{h \in F} |R(h) - R_n(h)|$, then we have a bound on the estimation error.
To do this, we will use some other mathematical results :

- Boole's inequality
- Hoeffding's inequality (non-asymptotical probabilistic bound)

Boole's inequality

Proposition

Let $A_1, A_2, \dots,$ *be acountable set of events of a probability space* $\{\Omega, \mathcal{F}, P)$.
Then,

$$P\Big(\cup_{i\geq 1} A_i\Big) \leq \sum_{i\geq 1} P(A_i) \tag{30}$$

This set might be infinite.

Boole's inequality

Proposition

Let $A_1, A_2, \dots,$ be acountable set of events of a probability space $\{\Omega, \mathcal{F}, P)$.
Then,

$$P\Big(\cup_{i \geq 1} A_i \Big) \leq \sum_{i \geq 1} P(A_i) \tag{31}$$

This set might be infinite.
Exercice 1 : Proove the proposition.

Hoeffding's inequality

Theorem

Hoeffding's inequality
Let $(X_i)_{1 \leq i \leq n}$ *be n i.i.d real random variables such that* $\forall i \in [1, n]$, $X_i \in [a, b]$ *and* $E(X_i) = \mu \in \mathbb{R}$. *Let* $\bar{\mu} = \frac{1}{n} \sum_{i=1}^{n} X_i$.
Then $\forall \epsilon > 0$,

$$P\Big(|\bar{\mu} - \mu| \geq \epsilon\Big) \leq 2 \exp\left(-\frac{2n\epsilon^2}{(b-a)^2}\right)$$

We admit this theorem.

Setting

- Supervised learning.
- Finite space of estimator F.
- The loss l is uniformly bounded : $l(\hat{y}, y) \in [a, b]$ with a and b real numbers.

Step 1

We have seen that

$$R(f_n) - R(f_a) \leq 2 \sup_{h \in F} |R(h) - R_n(h)| \tag{32}$$

As a consequence, for all $t \geq 0$:

$$P\Big(R(f_n) - R(f_a) \geq t\Big) \leq P\Big(2 \sup_{h \in F} |R(h) - R_n(h)| \geq t\Big) \tag{33}$$

Step 2

The fact that

$$2 \sup_{h \in F} |R(h) - R_n(h)| \geq t \tag{34}$$

is equivalent to :

$$\cup_{h \in F} \Big(2 |R(h) - R_n(h)| \geq t \Big) \tag{35}$$

Conclusion

Exercice 2 : Using Boole's inequality and Hoeffding's inequality, show that

$$P\Big(R(f_n) - R(f_a) \geq t\Big) \leq 2|F| \exp\Big(-\frac{nt^2}{2(b-a)^2}\Big) \qquad (36)$$

Conclusion

We write

$$\delta = 2|F| \exp\left(- \frac{nt^2}{2(b-a)^2} \right) \tag{37}$$

Exercice 3 :
We assume that $b - a = 1$. Show that with probability $\geq 1 - \delta$,

$$R(f_n) \leq R(f_a) + 2\sqrt{\frac{\log(|F|) + \log(\frac{2}{\delta})}{2n}} \tag{38}$$

Generalization

It is possible to generalize to infinite sets :

- by sampling F
- by using Rademacher complexity and Vapnik Vapnik-Chervonenkis theory.

This classical bound on the statistical error does not guarantee that the generalization error is small when $\log(|F|)$ is large.

Interpolation regime

- If the number of parameters is sufficient, it is possible to have $R_n(f_n) = 0$.
- In that case, it seems that the statistical error might to be way smaller than the previous bound.

For instance, for Wide Resnet, $\frac{p}{n} = 179$ with

- p : number of parameters in the netwok
- n : number of samples

Double descent

[Belkin et al., 2019]

Ensemble learning
- Bagging
- Random forest
- Boosting

Statistical learning
- Bounding the estimation error
- Interpolation regime and double descent

Model selection and sparsity
- Model selection
- Lasso

Example

- If $d >> n$ and we want to learn a linear model $x \mapsto \langle \theta, x \rangle$, we have seen that this raises statistical issues (high variance, overfitting).
- However, if we know in advance that θ only has $s < d$ non-zero coordinates (sparse θ), we can reformulate to an easier problem.
- But most of the time this is not the case, s is not known, so we need to test several subsets of non-zero coordinates.

Example

We could write the following regularized optimization problem

$$\hat{\theta} = \underset{\theta \in \mathbb{R}^d}{\arg\min} \left(||Y - X\theta|| + \lambda ||\theta||_0 \right) \tag{39}$$

- $y \in \mathbb{R}^n$ (labels)
- $X \in \mathbb{R}^{n,d}$ (design matrix)
- $||\theta||_0$: number of non-zero components of θ

Example

We could write the following regularized optimization problem

$$\hat{\theta} = \arg\min_{\theta \in \mathbb{R}^d} \Big(||Y - X\theta|| + \lambda ||\theta||_0 \Big) \tag{40}$$

- $y \in \mathbb{R}^n$ (labels)
- $X \in \mathbb{R}^{n,d}$ (design matrix)
- $||\theta||_0$: number of non-zero components of θ

However,

- optimization issue (not convex)
- computationally prohibitive to test all subsets of $[1, d]$.
  Exercice 4 : How many subsets does $[1, d]$ contain ?

Lasso

Le Lasso replaces $||\theta||_0$ by $||\theta||_1$.

$$||\theta_1|| = \sum_{i=1}^{d} |\theta_i| \tag{41}$$

Lasso estimator :

$$\tilde{\theta}_\lambda \in \underset{\theta \in \mathbb{R}^d}{\arg\min}\{||Y - X\theta||^2 + \lambda||\theta||_1\} \tag{42}$$

For technically involved reasons, the optimization with the lasso leads to sparser solutions in some situations.

Lasso

Lasso estimator :

$$\tilde{\theta}_\lambda \in \underset{\theta \in \mathbb{R}^d}{\arg\min} \{||Y - X\theta||^2 + \lambda ||\theta||_1\} \tag{43}$$

For technically involved reasons, the optimization with the lasso leads to sparser solutions. Frequently used optimization algorithm :

- coordinate descent (algorithm used in scikit)
- Fista
- LARS

https://en.wikipedia.org/wiki/Coordinate_descent

Example in 1D

Exercice 5 : What is the solution to the following optimization problem ?

$$\min_{\theta} F(\theta) = \frac{1}{2}(y - \theta)^2 + \lambda|\theta| \tag{44}$$

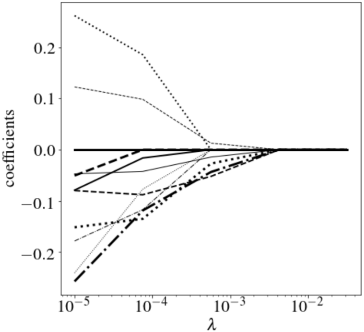


Figure – Regularization path with a Lasso optimization of a problem with $d = 12$. Each line represents the evolution of a θ_i when λ increases. Image from [Azencott, 2022].

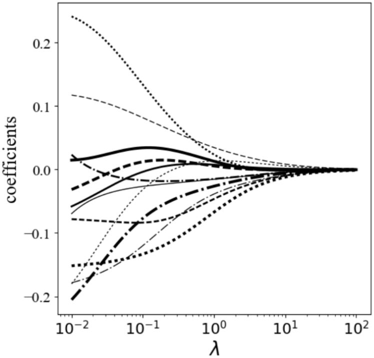


Figure – Regularization path with a Ridge optimization of a problem with $d = 12$. Each line represents the evolution of a θ_i when λ increases. Image from [Azencott, 2022].

Elastic net

Combination of $L1$ and $L2$ regularization.
Elastic-net estimator :

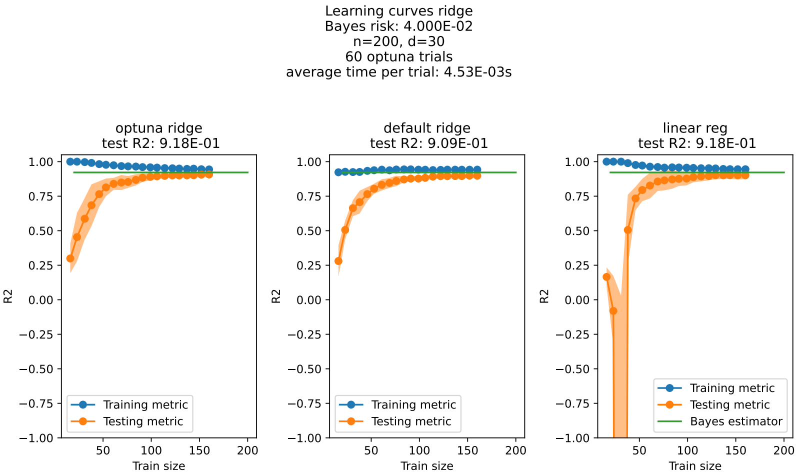
$$\tilde{\theta}_\lambda \in \underset{\theta \in \mathbb{R}^d}{\arg\min} \{ ||Y - X\theta||^2 + \lambda_1 ||\theta||_1 + \lambda_2 ||\theta||_2 \} \tag{45}$$

To choose λ_1 and λ_2 : cross validation.

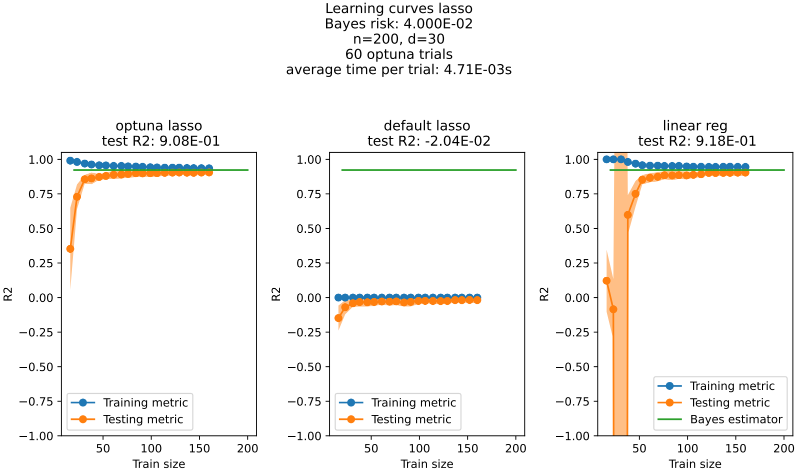
In the following examples, the data are linear, with a sparse θ^*.

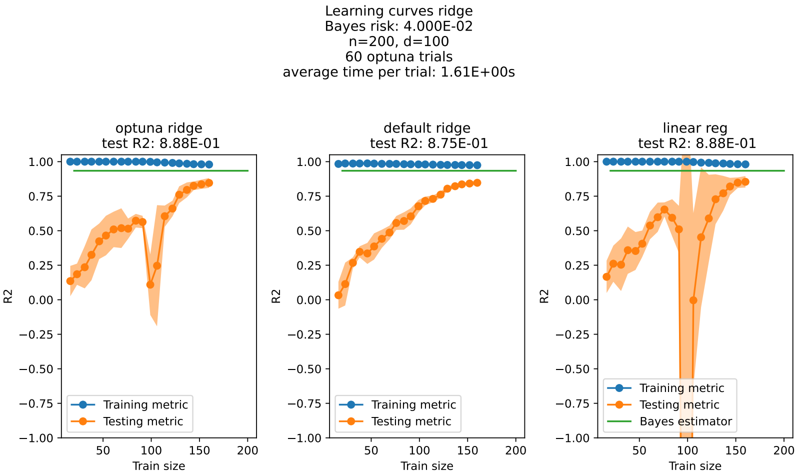
$$y_i = x^T\theta^* + \epsilon_i \tag{46}$$

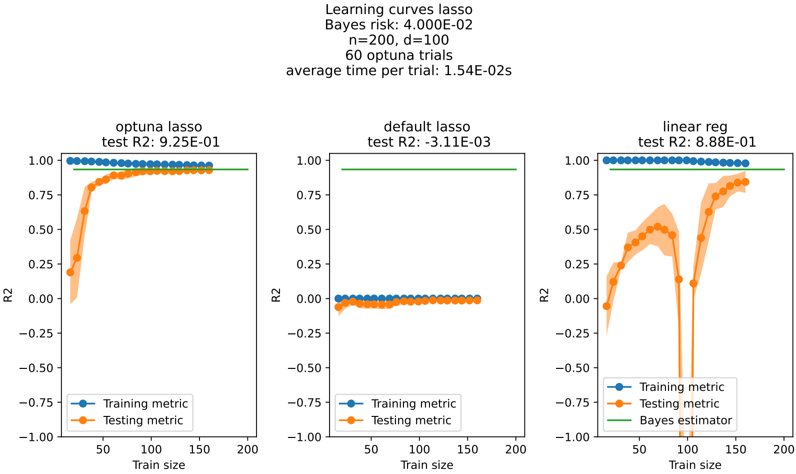
We compare Ridge and Lasso, for several dimensions (n, d).

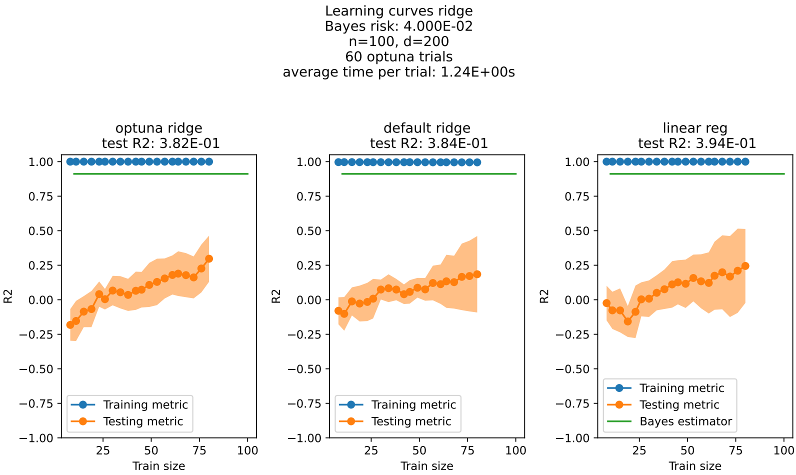


Learning curves ridge
Bayes risk: 4.000E-02
n=200, d=30
60 optuna trials
average time per trial: 4.53E-03s

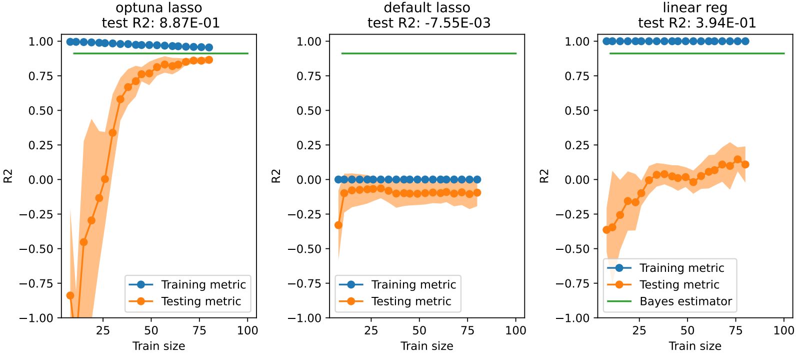

Learning curves lasso
Bayes risk: 4.000E-02
n=200, d=30
60 optuna trials
average time per trial: 4.71E-03s
optuna lasso
test R2: 9.08E-01
default lasso
test R2: -2.04E-02
linear reg
test R2: 9.18E-01
R2
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
50
100
150
200
Train size
Training metric
Testing metric
Bayes estimator


Learning curves ridge
Bayes risk: 4.000E-02
n=200, d=100
60 optuna trials
average time per trial: 1.61E+00s
optuna ridge
test R2: 8.88E-01
default ridge
test R2: 8.75E-01
linear reg
test R2: 8.88E-01
R2
Train size
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
50
100
150
200
Training metric
Testing metric
Bayes estimator

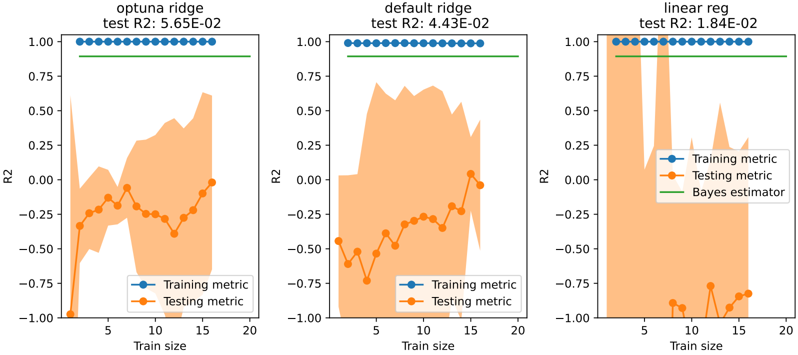

Learning curves lasso
Bayes risk: 4.000E-02
n=200, d=100
60 optuna trials
average time per trial: 1.54E-02s
optuna lasso
test R2: 9.25E-01
default lasso
test R2: -3.11E-03
linear reg
test R2: 8.88E-01
R2
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
50
100
150
200
Train size
Training metric
Testing metric
Bayes estimator

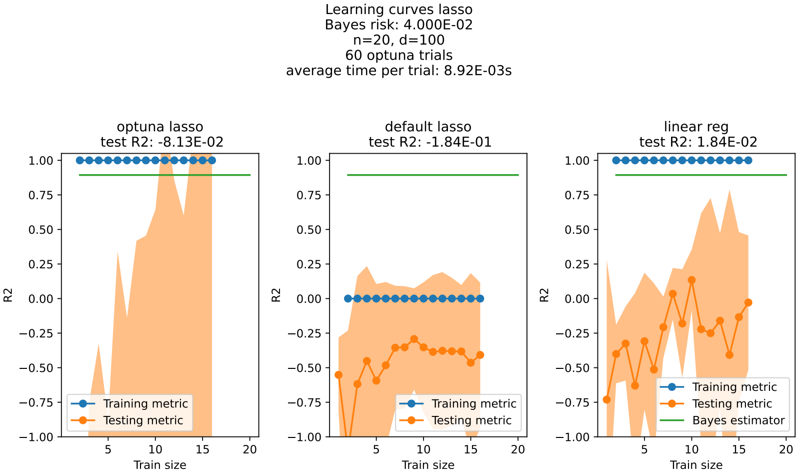

Learning curves ridge
Bayes risk: 4.000E-02
n=100, d=200
60 optuna trials
average time per trial: 1.24E+00s
optuna ridge
test R2: 3.82E-01
default ridge
test R2: 3.84E-01
linear reg
test R2: 3.94E-01
R2
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
25
50
75
100
Train size
Training metric
Testing metric
Bayes estimator

Learning curves lasso
Bayes risk: 4.000E-02
n=100, d=200
60 optuna trials
average time per trial: 1.77E-02s



Learning curves ridge
Bayes risk: 4.000E-02
n=20, d=100
60 optuna trials
average time per trial: 4.70E-03s





Learning curves lasso
Bayes risk: 4.000E-02
n=20, d=100
60 optuna trials
average time per trial: 8.92E-03s

optuna lasso
test R2: -8.13E-02

default lasso
test R2: -1.84E-01

linear reg
test R2: 1.84E-02

Multiple objective optimization

A estimator might be considered good for several reasons :

- quality of the predictions
- short(er) optimization time
- small(er) computational ressources

Multiple-objective optimization

In optuna for instance, it is possible to do **mutliple objective** optimization.

https://en.wikipedia.org/wiki/Pareto_front

https://optuna.readthedocs.io/en/stable/tutorial/20_recipes/002_multi_objective.html

https://optuna.readthedocs.io/en/stable/reference/visualization/generated/optuna.visualization.plot_pareto_front.html

References I

Azencott, C.-A. (2022).
Introduction au Machine Learning - 2e éd.

Belkin, M., Hsu, D., Ma, S., and Mandal, S. (2019).
Reconciling modern machine-learning practice and the classical bias–variance trade-off.
Proceedings of the National Academy of Sciences of the United States of America, 116(32) :15849–15854.

Breiman, L. (2001).
Random Forests.
Machine Learning, 45(1) :5–32.

References II

- Chen, T. and Guestrin, C. (2016).
XGBoost : A scalable tree boosting system.
Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 13-17-Augu :785–794.

- Fernández-Delgado, M., Cernadas, E., Barro, S., and Amorim, D. (2014).
Do we need hundreds of classifiers to solve real world classification problems ?
Journal of Machine Learning Research, 15 :3133–3181.

- Freund, Y. and Schapire, R. E. (1996).
Experiments with a New Boosting Algorithm.
Proceedings of the 13th International Conference on Machine Learning, pages 148–156.

References III

📄 Schapire, R. E. (2003).
The Boosting Approach to Machine Learning : An Overview BT - Nonlinear Estimation and Classification.
Nonlinear Estimation and Classification, 171(Chapter 9) :149–171.